ChinaXiv合作期刊

NLOF：基于网格过滤的两阶段离群点检测算法 *

王立英 [1,2]，石 磊 [2,3†]，伊 静 [1,4]，宋天霞 [1,2]

(1. 山东师范大学 信息科学与工程学院，济南 250014; 2. 山东省分布式计算机软件新技术重点实验室，济南 250014; 3. 山东教育招生考试院，济南 250014; 4. 山东建筑大学 计算机学院，济南 250014)

摘 要：离群点检测旨在有效识别数据集中的异常数据，挖掘出数据集中有意义的潜在信息。现有的离群度检测算法因没有对原数据进行处理导致计算时间复杂度过高，检测效果不理想。提出一种基于网格过滤的两阶段离群点检测算法 NLOF：首先使用网格过滤对原数据进行初步筛选，将密度小于特定阈值的数据放入候选异常子集中；然后为了进一步优化基于密度的算法，基于 k 邻域，根据邻域中数据点的个数与邻域所组成圆的面积之比，作为数据点密度计算的依据，进行离群点检测以获得更准确的离群点集。在多种公开数据集上进行实验，实验表明，该方法可以在异常检测中取得良好的性能，同时降低了算法的时间复杂度。

关键词：异常检测；网格过滤；局部密度；NLOF 算法

中图分类号：TP391 doi: 10.19734/j.issn.1001-3695.2018.09.0745

NLOF: two-stage outlier detection algorithm based on grid filtering

Wang Liying[1, 2], Shi Lei[2, 3†], Yi Jing[1, 4], Song Tianxia[1, 2],

(1. *School of Information Science & Engineering*, *Shandong Normal University*, *Jinan* 250014, *China*; 2. *Shandong Key Laboratory of Distributed Computer Software*, *Jinan* 250014, *China*; 3. *Shandong Education Admissions Examination Institute*, *Jinan* 250014, *China*; 4. *School of Computer*, *Shandong Jianzhu University*, *Jinan* 250014, *China*)

Abstract: The purpose of outlier detection is to effectively identify anomalous data in the dataset and to mine meaningful potential information in the data set. The existing outlier detection algorithm does not process the original data, resulting in too high computational time complexity and unsatisfactory detection results. This paper proposes a two-stage outlier detection algorithm NLOF based on grid filtering: First use grid filtering to initially screen the original data, put data with a density less than a certain threshold into a candidate exception subset; then in order to further optimize the density-based algorithm, based on the k-neighborhood, according to the ratio of the number of data points in the neighborhood to the area of the circle formed by the neighborhood, as the basis for calculating the data point density, outlier detection to obtain a more accurate outlier set. Experiments have been carried out on a variety of public datasets. Experiments show that this method can achieve good performance in anomaly detection and reduce the time complexity of the algorithm.

Key words: outlier detection; mesh filtering; local density; NLOF algorithm

chinaXiv:201901.00183v1

0 引言

离群点检测是数据挖掘领域的一个重要研究方向,用于在大量复杂的数据集合中消除噪声数据或者发现潜在未知的有意义信息。离群点检测可以广泛的应用在电子商务犯罪、信用卡欺诈侦查、网络入侵检测、生态系统失调检测、公共卫生、天文学未知种类天体发现等很多领域[1~3]。随着机器学习、模式识别、人工智能等领域的发展，越来越多有效的离群点检测方法不断地被人们提出。离群点检测算法有很多[4]，大致包括基于分布的方法、基于深度的方法、基于距离的方法、基于密度的方法等。最近还提出了一种基于海量数据聚类的高效异常检测方法[5] 。

每种类型的离群点检测算法都有其各自的优缺点，在基于分布的方法中，偏离标准分布的被认为是离群点，基于分布的方法可以简单有效地检测出异常值。但是基于分布的方法不适合分布未知的数据集。基于深度的方法可以解决这个问题，它假设数据在空间中由内到外一层一层包裹而成，越处在外层多边形上的数据点异常度越高，但这种方法对超过三维的数据就不甚理想。基于距离的方法[6]因为其有效性和简化性被广泛应用，但是基于距离的算法没有考虑局部密度的变化。基于密度的方法可以很好地解决这个问题。已经提出了很多基于密度的离群点检测算法，如 LOF[7]、INFLO[8]和 INS[9]。

基于密度的离群点检测算法也存在很大的缺点。算法通过计算数据集中每个数据点的离群因子值确定异常子集，通常选取离群因子值较大的若干个数据点作为离群点。这种利用离群因子来确定异常子集的方法在已知离群点数量的小规模数据集中检测效率很高，但是多数的离群点检测算法对离群点的个数未知，而且数据规模较大，所以对数据的过滤以减少正常数据的干扰显得格外重要。LOF 离群点检测算法

收稿日期：2018-09-29；**修回日期：**2018-11-19 **基金项目：**国家自然基金科学基金资助项目（61373148）；国家青年自然科学基金资助项目（61502151）；山东省社科规划项目（17CHLJ18，17CHLJ33，17CHLJ30）；山东省自然科学基金资助项目（ZR2014FL010）；山东省教育厅基金资助项目（J15LN34）

作者简介：王立英（1994-），女，山东济南人，硕士，主要研究方向为数据挖掘；石磊（1970-），男（通信作者），研究员，主要研究方向为网络模型及网络环境下应用技术研究（1542279330@qq.com）；伊静（1979-），女，博士研究生，副教授，主要研究方向为计算机网络、数据挖掘；宋天霞（1994-），女，硕士，主要研究方向为异常行为检测.

ChinaXiv合作期刊

需要重复计算 k 距离获得局部离群因子值，导致算法产生较大的时间复杂度，对 LOF 算法进行相应的改进也显得非常重要。

1　相关介绍

1.1　离群点定义

离群点[10]的定义根据不同的检测方法存在差别，但 Hawkins[11]给出了离群点的经典定义：一个离群点是一个数据点，它严重偏离其他数据点以至于怀疑是由不同机制生成的。被研究者广泛接受并成为研究离群点问题的基础。

1.2　基于网格的聚类方法

基于网格的聚类算法将空间量化为有限数目的单元，形成一个网格结构，所有聚类[12,13]都在网格上进行。基于网格的聚类方法采用空间驱动的方法，把嵌入空间划分成独立于输入对象分布的单元。基于网格的聚类方法使用一种多分辨率的网络数据结构。将对象空间量化成有限数目的单元，这些网格形成了网格结构，所有的聚类结构都在这种结构上进行。此方法的主要优点是处理速度快，其处理时间独立于数据对象数，而仅依赖于量化空间中的每一维的单元数。基于网格聚类的方法有很多，如 STING 算法[14]、Wave Cluster 算法[15]、CLIQUE 算法等。CLIQUE 算法是基于网格的空间聚类算法，它同时也非常好地结合了基于密度的聚类算法，因此既能够发现任意形状的簇，又可以像基于网格的算法一样处理较大的多维数据。

CLIQUE 算法[16~18]把每个维划分成不重叠的社区，从而把数据对象整个嵌入空间划分成单元。CLIQUE 算法把多维数据空间分割成若干个网格单元，将落到某个单元中点的数目当成这个单元的数据对象密度，指定一个阈值，当某个单元中点的个数大于这个阈值时，就说这是一个稠密单元格。CLIQUE 聚类算法是综合了基于密度和网格聚类算法的精华，处理大型数据库中混合类型及高维的空间数据，具有很高的效率，能够得到良好的聚类结果。

1.3　局部离群点检测算法 LOF

LOF 是 21 世纪初 Breuing 等人提出的一种基于密度的离群点检测算法。通过将某个对象的局部密度和周围密度相比得到数据对象的离群程度，可以较好地实现局部离群点的检测，在医疗处理、公共安全、工业损毁检测、图像处理和入侵检测等领域得到了广泛应用。因为 LOF 对密度是通过点的第 k 邻域来计算，而不是全局计算，所以得名为“局部”异常因子。

基于密度的离群点检测算法[21]是一种非常经典的异常数据挖掘算法[19,20]，主要通过比较每个点 p 和其邻域点的密度判断该点是否为离群点，如果点 p 的密度越低越可能被认定是离群点。密度的计算通过点之间的距离得到，点之间距离越远，密度越低，距离越近，密度越高。涉及到的有以下几个定义：

a) $d(p,o)$：两点 p 与 o 之间的距离。

b)k-distance：第 k 距离，p 的第 k 距离，也就是距离 p 第 k 远的点的距离不包括 p。

c)k-distance neighborhood of p：第 k 距离邻域。

点 p 的第 k 距离邻域 $N_k(p)$ 就是 p 的第 k 距离即以内的所有点，包括第 k 距离。因此 p 的第 k 邻域点的个数 $N_k(p)>=k$ 。

d) reach-distance：可达距离，点 o 到点 p 的第 k 可达距离定义为

$$reach-distance_k(p,o)=\max\{k-distance(o),d(p,o)\} \tag{1}$$

即点 o 到点 p 的第 k 可达距离，至少是 o 的第 k 距离，或者为 o p 间的真实距离。

e)local reachability density：局部可达密度。点 p 的局部可达密度表示为

$$lrd_k(p)=1/(\frac{\sum_{o\in N_k(p)}reach-dist_k(p,o)}{|N_k(p)|}) \tag{2}$$

f)local outlier factor：局部离群因子。点 p 的局部离群因子表示为

$$LOF_k(p)=\frac{\sum_{o\in N_k(p)}\frac{lrd(o)}{lrd(p)}}{|N_k(p)|}=\frac{\sum_{o\in N_k(p)}lrd_k(o)}{|N_k(p)|}/lrd_k(p) \tag{3}$$

LOF 离群点检测算法在实际应用中存在缺陷：LOF 算法需要计算数据点的离群因子值，数据集中的多数数据是正常点，离群点仅占据其中的小部分，这意味着 LOF 算法中存在大量无效的离群因子值计算，这种无效计算直接导致算法的更高时间复杂度。因此，本文提出了一种基于网格过滤的两阶段离群点检测算法。

2　本文算法

针对短时间内过滤数据库中的离群点准确率不高的问题，提出了本文算法。算法通过使用网格过滤器初步判断异常点，确保处理大型数据库中高维空间数据具有很高的效率，保证非稠密网格单元能够出现在候选离群点之内，高效减少数据量。通过改进的基于密度的离群点检测方法，减少算法的运行时间，提高算法效率，避免算法因为重复计算数据点的局部离群因子值产生的较大时间复杂度。下面是本文方法的相关介绍。

2.1　模型概述

两阶段的离群点检测算法主要包括两个步骤：

a）通过网格过滤完成对稠密网格单元的处理，确保非稠密网格单元能够出现在候选离群点之内，生成初步候选离群因子集。

b）通过改进的基于密度的离群点检测方法进行精确离群子集的判断。

由于基于网格过滤方法处理大型数据库中混合类型及高维的空间数据具有很高的效率，基于距离或密度的异常检测方法适合于局部异常检测，但是后者的检测精确度高于前者，所以对算法进行相应的改进，使其既包含处理高维数据的效率又包含局部离群点检测的精度。

基于网格的过滤算法考虑根据数据点所在网格的密度阈值判断数据是否异常，而此处的异常是一个全局的概念，很多局部异常的异常数据不会超过该密度阈值，所以将该算法作为一个过滤器，以密度阈值为判断依据，将密度小于阈值的判定为异常候选子集，在此过滤阶段会一定程度上减少传递给下一个算法的计算量。通过结合改进的基于密度的离群点检测方法，得到更为精确的异常数据集。

两阶段的离群点检测算法的大致流程如下所示。

```
High-level of the algorithm
a)//粗过滤阶段
Input Dataset
//输入数据集
Grid-based filtering
//利用网格过滤算法对数据进行初步筛选
Calculate  m  β
//通过 m  β 划分网格大小  密度阈值
Hash(把数据对象映射为 hash 表中的单元)
```

chinaXiv:201901.00183v1

ChinaXiv合作期刊


```
Getnumberpoints（）
//获得各个网格中的数据对象
if（Getnumberpoints（）< β）
//执行网格过滤采用递归判断的方法
Get Candidate Outlier Dataset;
//输出候选异常子集
b)//检测阶段
Input Candidate Outlier Dataset
//输入候选异常子集
Yi=Compute lrd_k(x_i)
//利用本文构造的离群因子计算数据点的离群度
if（Yi>di）
//执行异常检测采用递归判断的方法
Get Outlier Dataset
//输出异常子集
```

2.2　网格过滤阶段

在网格过滤阶段，此阶段主要负责进行数据集的读入和构建动态哈希表。将数据集存储到物理内存之中，当对读入的数据集进行扫描时，能够完成所有数据点映射任务，每个维度的数据点都会映射到与之相对的网格单元之内，通过借助哈希表储存结果，实现对输入数据信息的动态化读入。此种算法可以实现对所有网格单元数据点数的实时计算，且可以对数据计算准确度进行保证，通过按照事先所设定的阈值完成对网格单元类型的判断工作，可以在最短时间内精准判断出网格单元的类型，并完成对网格单元的处理，以保证算法有效性。

网格过滤阶段对数据集进行基于网格的映射，通过排除各个密集簇，得到初步的异常子集。采用网格划分的方法，将数据集合划分为网格，每一个单元格代表一个子簇，以单元格中的对象个数为近似密度。网格过滤阶段，算法需要两个参数，一个是网格的步长，第二个是密度的阈值。网格步长确定了空间的划分，而密度阈值用来定义密集网格。

网格过滤阶段，算法需要的第一个参数是网格步长，设数据集 $N=\{o_1,o_2,\ldots,o_j,\ldots o_N\}$ 中的数据点分属于各个网格之中，网格的数量大小为 $m*m$,网格划分的大小和数据集的大小相互制约。当数据集较大时，网格的大小应相对较大，否则数据分布较为密集，无法正常区分候选异常子集。当数据集较小时，网格划分大小应相对较小；否则数据分散较为稀疏，无法正常区分以排除各个密集簇。本文提出一个与数据集的大小相关的网格划分数量函数，使得满足上述条件。通过实验验证，网格划分大小和数据集数量存在函数关系，通过总结各个数据集大小与网格步长的关系，提出本文的网格划分函数：

$$m=[|N|^{\frac{1}{3}}+|N|^{\frac{1}{4}}] \tag{4}$$

其中：|N| 为数据集 N 的数量大小；m 为网格的行数（列数）。

网格过滤阶段，算法需要的第二个参数是密度阈值，密度阈值用来定义密集网格。根据数据集的大小和网格数量的关系，网格数量随数据集大小的变化而改变。密度阈值的确定又与数据集大小、网格步长相关，所以密度阈值和数据集的大小息息相关。当数据集较大时，网格中数据点的分布会相应的增加，此时密度阈值应相应的增大；否则，无法确保非稠密网格单元能够出现在候选离群点之内。当数据集较小时，网格中数据点的分布会相应减少，此时密度阈值应相应减少；否则，部分稠密网格单元出现在候选离群点之内，使得网格过滤失去自身的过滤效果。所以在网格过滤阶段提出一个和数据集大小相关的密度阈值函数。

通过实验验证不同数据集大小与密度阈值之间的关系，总结各个数据集大小与密度阈值之间的关系，提出本文的密度阈值函数：

$$\beta=[\frac{|N|^{\frac{1}{3}}+|N|^{\frac{1}{4}}}{3}] \tag{5}$$

其中：β 为网格过滤的密度阈值；|N| 为数据集 N 的数量大小。

网格过滤阶段，算法需要的第一个参数是网格步长，算法需要的第二个参数是密度阈值。为了验证本文提出的网格划分函数和密度阈值函数的合理性，使用人工合成的数据集进行测试。表 1 所示为数据规模包括 N=100,N=1000,N=2000,N=5000,N=10000,N=20000,N=50000, N=100000 的数据集合，经本文 m 和 β 函数，过滤前后数据样本的变化结果。

表 1　样本变化统计

Table 1　Sample change statistics

样本个数	异常点数量	m	β	过滤后的样本个数	过滤后异常点个数
100	4	8	3	15	4
1000	9	16	5	32	9
2000	15	19	6	42	15
5000	31	26	9	113	31
10000	68	31	11	161	68
20000	113	39	13	173	113
50000	196	51	17	497	196
100000	365	64	21	9724	365

过滤阶段通过给定的数据集，根据本文提出的网格划分函数、密度阈值函数，计算网格步长和合理的密度阈值，根据密度阈值判断网格是稠密网格或非稠密网格。若网格为非稠密网格，则将该网格中的数据点放入异常数据候选子集。

因为异常具有数量少且与正常数据不同的特点，所以分布相对稀疏，密度系数较小；当密度系数 y 小于密度阈值时，将相应的数据放入候选异常子集。因为将每个对象指派到一个单元并计算每个单元密度的时间复杂度和空间复杂度为 O(n)，所以整个网格过滤过程是非常高效的。最后再进行改进的基于密度的离群点检测方法进行精确检测，减少算法运行时间，提高算法运行效率。

2.3　精确检测阶段

通过上一阶段的过滤，可以得到初步异常子集，再通过更加精确的离群点检测算法检测。基于密度的离群点检测算法是一种非常经典的异常数据挖掘算法，主要通过比较每个点 p 和其邻域点的密度来判断该点是否为离群点。如果点 p 的密度越低，则越可能被认定是离群点。至于密度，是通过点之间的距离来计算的，点之间距离越远，密度越低，距离越近，密度越高。用 lrd 表示局部可达密度。点 p 的局部可达密度表示为

$$lrd_k(p)=\frac{\Sigma_{o\in N_k(p)}reach-dist_k(p,o)}{|N_k(p)|} \tag{6}$$

用 local outlier factor 表示局部离群因子。点 p 的局部离群因子表示为

$$LOF_k(p)=\frac{\Sigma_{o\in N_k(p)}lrd_k(o)}{|N_k(p)|*lrd_k(p)} \tag{7}$$

$LOF_K(P)$ 的值越接近 1，说明 p 的邻域点密度相差不大，p 可能和邻域同属一簇；比值越小于 1，说明 p 的密度高于其邻域点密度，p 为密集点；如果这个比值越大于 1，说明 p 的密度小于其邻域点密度，p 越可能是离群点。在计算点

chinaXiv:201901.00183v1

ChinaXiv合作期刊

o 到点 p 的第 k 可达距离时，时间复杂度为 $O(N_k^2)$。

在基于网格的过滤算法中，密度表示单位面积内点的个数，单位面积内点的个数越多，则密度越大。单位面积内点的个数越少，则密度越小。换句话说，数据点分布的范围越小，则说明数据之间的距离越紧密，数据是正常点的可能性越大。数据点分布的范围越大，则说明数据之间的距离越稀疏，数据点是离群点的可能性越大。

如图 1 所示，可达距离 k=3 时数据的分布，当以 k=3 为圆的半径时，p1 点的邻域所组成的单位圆面积相比较于 P2 点的邻域所组成的单位圆面积大很多。本文将第 k 距离邻域中数据点的个数，与所组成的圆面积的比值，作为数据点密度的判断依据，算法的时间复杂度为 $O(N_k)$，相较于其他离群点检测算法大大减少了算法所需要的时间，提高了离群点检测的算法效率。

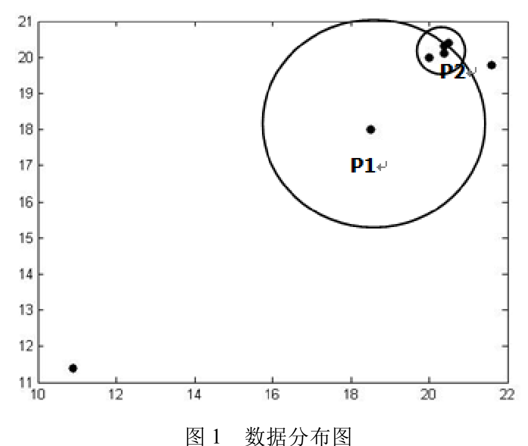


图 1　数据分布图

Fig. 1　Data distribution map

改进的基于密度的局部离群点检测算法基于异常数据的分布通常比正常簇分布稀疏的多的想法。根据第 k 距离邻域中数据点的个数，与第 k 距离邻域所组成的圆的面积之比，作为数据点密度的判断依据。提出本文改进的基于密度的离群点检测算法：

$$lrd_k(p) = N_k(p) / \pi * (k - distance(p))^2 \tag{8}$$

数据点的局部密度越小，则数据分布在 k 距离邻域的范围越大，数据点是离群点的可能性越大；同理，数据点的局部可达密度越大，则数据分布在 k 距离邻域的范围越小，数据点是离群点的可能性越小。

2.4　算法步骤

a)输入密度阈值，应用网格过滤算法，得到初步的离群点集，此处的密度阈值通过密度阈值函数得到，再通过下一步进行更为精确的优化

b)对离群点集进行改进的基于密度的异常检测，得到最终的离群点集。

```
a)//粗过滤阶段
Grid-based filtering
Initialize:new cell()//初始化，网格标记为0
Read()
Get points(x,y,z)
Reflect point（cellx，celly，cellz）
Addnumberpoints()//对应网格计数加1
if（getnumberpoints>=densityThreshold）
{
SetQuatified(1)
//网格数量大于阈值设置网格标记为1
}
Get pointstr1(x,y,z)
Reflect point（cellx，celly，cellz）
if(GetQuatified(1))
{
Printf(pointstr1)
//获得小于密度阈值的所有数据点
}
b)//检测阶段
Yi=Compute  lrd_k(x_i))
if Yi>di
Output：Outlier Dataset
```

3　实验及结果分析

3.1　数据集

为了验证本文提出的基于网格过滤的两阶段离群点检测算法的性能，对比本文算法和 LOF 离群点检测算法，通过合成数据集和实际数据集的运算结果来检测算法的有效性和可用性。为了比较算法在不同数据规模的检测性能，使用人工合成的数据集进行测试，数据规模包括 100~100000 间的数据集合、选取来自 UCI 标准数据库的实验数据进行验证。因为离群点在数据集中的数量较少，所以为了符合数据集的分布规律,将数据集中的某一类的部分对象删除并作为离群点。算法在 VS2010 和 matlabR2014a 中实现。在 12 核的 Intel Xeon 3.5 GHz CPU，32 GB RAM 的 64 位的 Windows10 平台进行实验。表 2 为相关数据集信息的描述。

表 2　实验数据统计

Table 2　Statistics of experimental data

数据集	样本个数	属性	类数
Aggregation	788	2	7
Compound	399	2	5
Heart	150	13	2
Liver disorder	150	6	2

通过网格过滤的算法对表 2 所示的数据集进行候选异常子集的筛选，k 距离邻域的大小对算法的性能有重要影响，如果选择的 k 太小，则可能无法检测到异常值簇；相反，可以将正常点检测为异常值。表 3 显示了当 k 取不同值时对 LOF 算法精确度的影响。当 k=8 时，LOF 算法的精确度最高，选取 k=8 为本文的 k 距离邻域。

表 3　k 值精确度

Table 3　K accuracy

数据集	K	LOF
Aggregation	7	69.21%
	8	69.39%
	9	67.98%
Compound	7	68.34%
	8	69.98%
	9	67.54%
Heart	7	70.36%
	8	70.36%
	9	68.11%
Liver disorder	7	63.56%
	8	62.31%
	9	61.76%

3.2　对比实验

为了验证本文提出的网格过滤算法的性能，图 2 为经网格过滤前后数据量的对比。基于网格过滤的主要目的是对数据集进行初步的筛选，即把数据集最稠密的数据簇去掉，并

chinaXiv:201901.00183v1

ChinaXiv合作期刊

尽可能的使待检测的数据量达到最小，大大减少算法运行时间。根据网格划分函数、密度阈值函数，对不同数据量的数据集进行网格大小和密度阈值的计算。通过图 2 发现，当数据量为 0~100000 时，网格过滤前后数据量发生了很大的改变。例如，当数据集为 100 000 时，经网格过滤，数据量变为 10 742 个，大大减少了数据集的数量，减少了算法所需要的时间，提高了算法效率。然后对待检测的数据利用改进的基于密度的离群点检测算法计算数据的离群率。

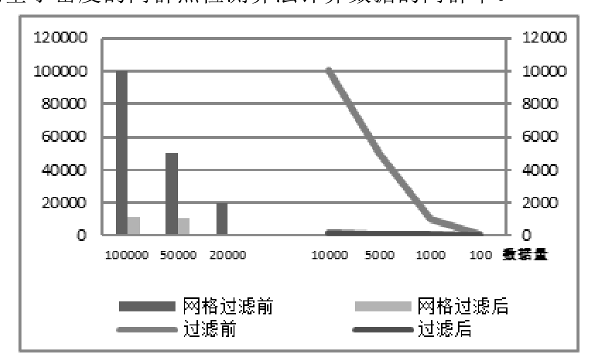


图 2 网格过滤前后数据量对比

Fig. 2 Comparison of data volume before and after grid filtering

通过改进的基于密度的离群点检测算法对候选异常子集进行更为精确的检测，将时间复杂度为 $O(N_k^2)$ 的局部离群点检测算法转换为时间复杂度为 $O(N_k)$ 的基于密度的局部离群点检测算法。为了验证本文提出的基于网格过滤的两阶段离群点检测算法所消耗时间的异常检测性能，图 3 显示了 NLOF 算法所消耗时间和 LOF 算法所消耗时间的对比。为了比较算法在不同数据规模的检测性能，使用多个数据规模的数据集合分别进行测试。通过对比不同规模的数据集，NLOF 算法的时间检测效果都非常出色，相较于原算法大大缩短了离群点检测所需要的时间。

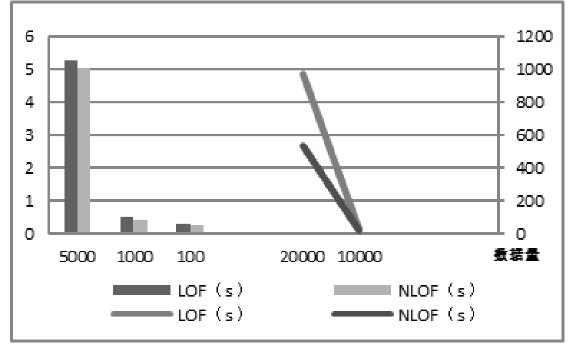


图 3 NLOF 算法消耗时间对比

Fig. 3 NLOF algorithm consumption time comparison chart

为了验证本文提出的基于网格过滤的两阶段离群点检测算法的性能，表 4 列出了通过对比合成数据集和真实数据集得到的结果。表 4 所示为输出的离群率最大的数据对象中，正确找到离群点所占总离群点数量的百分比。通过网格过滤对数据集进行初步筛选，形成候选异常子集，再对候选异常子集进行更加精确的离群点检测，大大减少了算法所消耗的时间。通过对比不同数据规模、不同离群点个数，对算法的精确度进行对比，NLOF 算法具有更精确的运算结果和更短的运行时间。

由于合成数据集是人为生成的数据集，具有一定的规律性，本文又对比了真实实验数据集，如表 4 所示的离群点检测实验结果 NLOF 算法仍具有更好的检测性能。通过对比四个数据集的不同样本数，NLOF 离群点检测算法较原 LOF 算法具有较高准确率。

表 4 实验结果

Table 4 Experimental results

数据集	样本个数	异常数据	本文算法	LOF 算法
Aggregation	788	24	89.93%	69.39%
Compound	399	18	91.02%	69.98%
Heart	150	15	92.53%	70.36%
Liver disorder	150	10	93.93%	62.31%

4 结束语

本文深入研究了现有的离群点检测算法的缺点，探讨了基于网格过滤和基于密度的离群点检测算法存在的优势和不足。本文提出的基于网格过滤的两阶段离群点检测算法能够很好地将处理高维数据的效率与局部检测的精度结合起来，在解决不同数据量和数据类型的问题上取得了比以往离群点检测算法更加优秀的结果。通过在不同的参数下与原检测方法进行比较，得到本文所提出的算法具有更好的检测效果。

参考文献：

[1] Jin Wen, Tung A K H, Han Jiawei,et al. Mining top-n local outliers in large databases [C]//Proc of the 7th ACM SIGKDD International Conference.San Francisco, California: ACM Press, 2001: 293-298.

[2] 王茜，刘书志. 基于密度的局部离群数据挖掘方法的改进 [J]. 计算机应用研究，2014, 31 (6): 1693-1696. (Wang Qian, Liu Shuzhi. Improvement of local outliers mining based on density [J]. Application Research of Computers, 2014, 31 (6): 1693-1696.)

[3] Han Jiawei, Micheline K. Data mining: concepts and techniques [J]. Data Mining Concepts Models Methods & Algorithms, 2006, 5 (4): 1-18.

[4] Huang Jinlong, Zhu Qingsheng, Yang Lijun, *et al.* A novel outlier cluster detection algorithm without top-n parameter [J]. Knowledge-Based Systems, 2017, 121: 32-40.

[5] Kim H, Min J K. An energy-efficient outlier detection based on data clustering in WSNs [J]. International Journal of Distributed Sensor Networks, 2014, 2014 (2): 1-7.

[6] Zhang Ke, Hutter Marcus, Jin Huidong. A new local distance-based outlier detection approach for scattered real-world data[C]//Advances in Knowledge Discovery and Data Mining. Berlin: Springer-Verlag, 2009. 813-822.

[7] Oliván A D, Pagán J A, Sanz R, *et al.* Data-driven prognostics using a combination of constrained K-means clustering, fuzzy modeling and LOF-based score [J]. Neurocomputing, 2017, 241.

[8] Jin Wen, Tung A K H, Han Jiawei, *et al.* Ranking outliers using symmetric neighborhood relationship [J]. Lecture Notes in Computer Science, 2006, 3918: 577-593.

[9] Ha J, Seok S, Lee J S. Robust outlier detection using the instability factor [J]. Knowledge-Based Systems, 2014, 63 (2): 15-23.

[10] Chiang A, David E, Lee Y J, *et al.* A study on anomaly detection ensembles [J]. Journal of Applied Logic, 2017, 21: 1-13.

[11] Hawkins D M. Identification of outliers [M]. London: Chapman and Hall, 1980.

[12] Ayed A B, Halima M B, Alimi A M. Survey on clustering methods: towards fuzzy clustering for big data [C]//Proc of IEEE Soft Computing and Pattern Recognition. 2015: 331-336.

[13] 田启明，王丽珍，尹群. 基于网格距离的聚类算法的设计、实现和应用 [J]. 计算机应用，2005, 25 (2): 294-296. (Tian Qiming, Wang Lizhen, Yin Qun. Design, realization and application of clustering

chinaXiv:201901.00183v1

ChinaXiv合作期刊

algorithm based on the distance between grids [J]. Computer Applications, 2005, 25 (2): 294-296.)

[14] Bureva V, Sotirova E, Popov S, *et al.* Generalized net of cluster analysis process using sting: a statistical information grid approach to spatial data mining [C]//Proc of International Conference on Flexible Query Answering Systems. 2017: 239-248.

[15] Sheikholeslami G, Chatterjee S, Zhang Aidong. WaveCluster: a multi-resolution clustering approach for very large spatial databases [C]//Proc of International Conference on Very Large Data Bases. 1998: 428-439.

[16] Sæther S H, Telle J A. Between treewidth and clique-width [J]. Algorithmica, 2016, 75 (1): 218-253.

[17] Xu Yanyan, Cheng James, Fu Ada WaiChee: distributed maximal clique computation and management [J]. IEEE Trans on Services Computing, 2016, 9 (1): 110-122.

[18] Malladi K T, Mitrovic-Minic S, Punnen A P. Clustered maximum weight clique problem: algorithms and empirical analysis [J]. Computers & Operations Research, 2017, 85.

[19] Wu Mingxi.Outlier detection by sampling with accuracy guarantees [C]//Proc of the 12th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.2006: 767-772.

[20] Tang Bo, He Haibo. A local density-based approach for outlier detection [M]. [S.l.]:Elsevier Science Publishers B. V, 2017.

[21] Su Shubin, Xiao Limin, Zhang Zhoujie, *et al.* N2DLOF: a new local density-based outlier detection approach for scattered data [C]//Proc of IEEE International Conference on High Performance Computing & Communications.2017: 458-465.

[22] Maratea A, Petrosino A, Manzo M. Adjusted F-measure and kernel scaling for imbalanced data learning [J]. Information Sciences, 2014, 257 (2): 331-341.

[23] Salehi M, Leckie C, Bezdek J C, *et al.* Fast memory efficient local outlier detection in data streams (extended abstract) [C]//Proc of IEEE, International Conference on Data Engineering. 2017: 51-57.

chinaXiv:201901.00183v1